An Evaluation of Web Plagiarism Detection Systems for Student Essays

Tuomo Kakkonen, Maxim Mozgovoy
Department of Computer Science and Statistics, University of Joensuu, Finland
tuomo.kakkonen@cs.joensuu.fi

Abstract: This study uses purpose-built test data and empirical experiments to report on the performance of four web plagiarism detection systems: TurnitIn, SafeAssignment, Plagiarism-Finder and EVE. In addition to measuring accuracy of detection, we evaluated the extent to which these systems produce false detections. We obtained the test data from multiple sources and edited it in several ways to conceal the plagiarisms in the texts. Our results indicate that SafeAssignment was the best overall performer.

Keywords: Web plagiarism, plagiarism detection, student essays, evaluation

1. Introduction

Plagiarism has become a serious problem in education. The ease with which anyone can copy and collate texts from the Internet, make *web plagiarism* deeply tempting to some students. A number of researchers suggest that the ease with which it is possible to plagiarize web sources with impunity has significantly increased the amount of plagiarism that students commit. It is this type of plagiarism with which we are concerned in this work.

Teachers and academics abhor plagiarism because it is inconsistent with pedagogical aims. The most common method of detecting plagiarism relies on the ability of an assessor to make deductions about the probability of plagiarism on the basis of internal clues embedded in the text itself. Countering plagiarism by using such “traditional” means is unfortunately ineffective. While Internet search engines, such as Google, can be used to detect Internet plagiarism, the detection process is, by any standards, both tedious and labour-intensive. It is obvious that such a process is extremely time-consuming. Fortunately, however, there are a number of companies, such as iParadigms [7] and Canexus [3], who have developed automatized plagiarism detection software.

In this paper, we introduce some of the most prominent of these automatic plagiarism detection systems in conjunction with the results of a study which we undertook to evaluate their ability to detect types of plagiarism ranging from direct copying to paraphrasing. The paper is organized in the following way. In Section 2 we emphasize the characteristics and features that any successful detection system needs to have. Section 3 describes our experiments with the detection systems and the results of those experiments.

2 Requirements for a successful plagiarism detector

An automatic plagiarism detector has to be able to identify two kinds of plagiarism: (1) the plagiarism involved in copying from another student’s work, and (2) the plagiarism involved in copying without acknowledgement from reference materials (such as those in textbooks and the Internet). There are in fact numerous styles of plagiarism. These range from direct copying to making use of a ghostwriter. A detection system should ideally be able to detect all instances and styles of plagiarism while refraining from the stigmatization of texts in which no plagiarism has in fact occurred. The following list reflects some of the most common types of plagiarism:

1. Copy-paste/verbatim copying and the word-for-word transcription of texts.
2. Paraphrasing: the reordering of sentences and effecting changes to grammar and style while inserting words with similar meanings (synonyms).
3. The deliberately inaccurate use of bogus references: the making of references to incorrect or non-existent sources.
4. The insertion of similar-looking characters from foreign alphabets. Thus, for example, the letter "O" can be equally well represented with the following three different characters: Unicode 004F (Latin O), 039F (Greek Omicron), and 041E (Cyrillic O).
5. The insertion of invisible white-colored letters into what seem to be blank spaces. Most modern text processors allow the user to specify a font color in a document. The plagiarism could exploit this feature by inserting a white font in a blank space with a white background. This would have the effect of distorting the content of the text even though, to the naked eye, it would be visually identical to the original.

A limitation associated with existing systems is the possibility of *false detections* (i.e. the return of false positives). Since the volume of texts on the Internet is so vast, it is possible for parts of a student's text coincidentally to resemble text from an existing web page – even though the student might never have seen the web page concerned. This kind of resemblance is often referred to as "casual similarity". Students may also cite materials that they themselves published on the Internet. It is, of course, perfectly legal to quote fragments from informally published self-authored text.

3 Experiments

The plagiarism detection systems we evaluated are introduced in Section 3.1. We performed our evaluation by first collecting a set of test documents that contained several types of plagiarism. The test data and our test settings are described in Section 3.2. The results of the experiments and the conclusions that we drew from them are described in Section 3.3.

3.1 Web plagiarism detection systems

SafeAssignment, one of *MyDropBox*'s tools [6], is a system that can perform both local and Internet detection. While the user interface is intuitive, the addition of new essays for scrutiny is complicated by the fact that these essays need to be submitted one by one. The system gives the user no control over the detection method that should be used.

TurnitIn [7], which claims that thousands of schools and universities around the world use its services, is arguably is the most widely used of all current plagiarism detection systems. The system produces an originality report on a student text by comparing it, not only to the pages and documents on the Internet, but also to its own essay database of more than 40 million student papers.

The *EVE2* ("Essay Verification Engine") system, which was developed by Canexus [3], keeps no database of its own essays or texts. Its method is to search the Internet for essays by making use of existing Web searching engines. We used version 2.5 of this system in our research. The program installs to the user's own computer and allows the user to select from three possible detection modes: *quick*, *medium* and *full strength*. Adding a file to the system is a cumbersome process because it only accepts up to ten files at once.

Plagiarism-Finder (version 1.3.0) [5] works in more or less the same way as EVE2. It installs onto the user's computer and searches the Internet for possible occurrences of text fragments from the local document collection. Plagiarism-Finder allows a user to adjust the detection algorithm by making adjustments to two parameters: the record length (in words) that needs to be checked, and the increment (in words) that defines the size of the step whereby advances are made to the next "record" (a sequence of words) in the document

3.2 Test set and settings

The construction of the test set consisted of three phases. In the first phase, we collected a set of sentences that belonged to the following three categories: original, web and mill. The sentences in the *Original* category were either deliberately written for the test set by the authors of this paper, or were sourced from books that had not already been published online. The *Web* sentences were obtained from a variety of Internet pages. The papers in the *Mill* category were acquired from "paper mill"[1] services such as [1] and [2].

The test sentences described in the previous paragraph constituted the *verbatim sentences*. We then modified these sentences in three different ways so that we would have groups of sentences that were *edited, synonymous* or *paraphrased*. The *edited* sentences were characterized by minor alterations such as added spaces, intentional spelling errors, deleted or added commas, and periods that were replaced by exclamation marks. In the *synonymous* sentences, one or two words from each sentence were replaced with exact or close synonyms. The *paraphrased* sentences were characterized by a wide range of sentence alterations of the following kind: they included the kind of alterations found in the edited and synonymous sentences, and, in addition, changes in the original order of words and phrases. Thus, for example, the sentence, "He ate pizza and pasta and she drank coffee", could be rearranged so that it read, "She drank coffee and he ate pasta and pizza".

The test set contained a total of 1,200 sentences: 100 sentences for each test sentence type. The sentences were sourced from texts in the following fields: the use of technology in education, natural languages, natural language parsing and understanding, automatic essay grading, artificial intelligence, and object-oriented programming. In the last phase, we collected the sentences into test files, with each file containing only sentences that represented a single test type. We also preserved the original order of the sentences so that each sequence of test sentences formed a coherent passage of text. This resulted in the 48 test files.

We carried out separate test runs for the files that belonged to each of the three categories (Original, Web, Mill). Each test run therefore consisted of 16 files. The tests were first run with the default settings of each of the detection systems. For those systems in which the user is given control over the detection mechanism, we ran the tests on the most stringent settings available. Table 1 summarizes the settings for each system.

Table 1. The test settings for each of the systems. The columns "Default" and "Strict" describe the default and strictness of the test settings for each of the systems respectively.

System	**Default**	**Strict**
SafeAssignment	Default	-
TurinitIn	Default	-
EVE	Normal	Full strength
Plagiarism-Finder	Normal, record length 7 words, increment 50 words	Detailed, record length 4 words, increment 2 words

[1] "Paper mills" are Internet services that offer student essays for free or for payment.

3.3 Results

Table 2. Evaluation results. The numbers indicate the percentage of the files correctly detected. In the Original category, for example, 100% indicates that none of the files contained false detections. KEY: SA = Safe-Assignment, PF = Plagiarism-Finder

		SA	*TurnitIn*	*EVE*		*PF*	
		Default	*Default*	*Default*	*Strict*	*Default*	*Strict*
Original	**Verbatim**	100,0	100,0	100,0	100,0	100,0	0,0
	Edited	100,0	100,0	100,0	100,0	100,0	0,0
	Synonymous	100,0	100,0	100,0	100,0	100,0	0,0
	Paraphrase	100,0	100,0	100,0	100,0	100,0	0,0
	TOTAL	100,0	100,0	100,0	100,0	100,0	0,0
Web	**Verbatim**	100,0	75,0	0,0	0,0	0,0	0,0
	Edited	100,0	75,0	0,0	0,0	0,0	0,0
	Synonymous	100,0	75,0	0,0	0,0	0,0	0,0
	Paraphrase	100,0	50,0	0,0	0,0	0,0	0,0
	TOTAL	100,0	68,8	0,0	0,0	0,0	0,0
Mill	**Verbatim**	100,0	50,0	0,0	0,0	0,0	0,0
	Edited	100,0	75,0	0,0	0,0	0,0	0,0
	Synonymous	0,0	50,0	0,0	0,0	0,0	0,0
	Paraphrase	50,0	25,0	0,0	0,0	0,0	0,0
	TOTAL	62,5	50,0	0,0	0,0	0,0	0,0

As shown in Table 2, in the Original texts, SafeAssignment identified between one and three sentences in some of the documents as sentences that had been copied from a web page. But because the overall plagiarism score (between 3 and 11) for the test document in each of these cases was smaller than the 25% limit set for the “yellow” category with a moderate plagiarism risk, we did not classify these cases as false detections. Table 3 shows that SafeAssignment’s detection accuracy was extremely good in the Web category. It returned a plagiarism score of less than 100% to only two of the 16 files in this category.

Even though TurnitIn did not produce any false detections from the Original data, its detection accuracy was poorer than that of SafeAssignment in all test categories except for the synonymous tests on the Mill data. While testing the Web texts, the system failed to detect a document that had been sourced from a web page on the .fi domain and therefore failed to produce accurate results for that document in all the test categories. This indicates the existence of a gap in its Internet search coverage. The percentage of files that the system identified as plagiarized on the Mill data was higher than the figures given in Table 2. In some cases, the source from which we had taken our test sentences was not among the sources indicated in TurnitIn’s output. Since our test requirement was a correct identification of each instance of plagiarism as well as its source, we did not accept these files as correctly identified. One advantage that TurnitIn has over the other systems is that it includes a collection of essays sourced from its clients as part of its database. Our test set generated plagiarism warnings from these essays, especially on the Mill data.

Although EVE returned no false identifications for the Original sentences with either the default or the strict setting, its detection accuracy was nevertheless terrible with both settings. It was not able to identify a single instance of plagiarism on our test set! In order to ensure that this performance had not been caused by the file format or the test settings, we carried out the detection runs with all three of the possible detection levels offered by the tool. Even though we also conducted the experiments with Plain Text and Microsoft Word documents, both of which EVE claims to support, its performance did not improve.

Plagiarism-Finder failed to impress us with its default settings. While it did not generate any false alarms with the Original test data, it was only able to identify low overlaps with Internet sources – even with the Web verbatim documents. These overlapping sections had, in most cases, not been obtained from the documents from which we had sourced the sentences. For one of the Web verbatim files, for instance, the system (with its default settings) identified 9% of the sampled words as sourced from the Internet, and a total of 1% of the essay as directly copied!

The fact that Plagiarism-Finder with the strict settings makes so many false detections and coincidental matches represents a serious defect. Its most common result was a set of three word fragments that matched different Internet sources. It stated, for example, that between 19 and 26 per cent of the words in the Original files were copied from the Internet. Surprisingly, the percentage of words that overlapped with Internet sources was only slightly higher (between 25 and 28%) on the Web verbatim category! Not only did its default settings generate a lack of accuracy, its strict settings also made the detection extremely noisy.

A closer inspection of Plagiarism-Finder's results leads to even more worrying observations. Almost 100% of the matches in the Web category were not to the sources from which we had taken the sentences. We also noticed that almost all the detections were made with text from web pages on the .de, .at and .ch domains or links to Excite.de [4] search engine. It thus appears that the system only searches a small German section of the Internet and uses of a web search engine with rather poor coverage.

4 Conclusion

The results of our experiments with two locally installable and two Internet-based web plagiarism detection systems revealed that while SafeAssignment was the detector that offered the best all-round performance, the two locally installable systems performed poorly. The only category in which SafeAssignment failed to reach 100% performance was the paper mill test set. This indicates that the internal essay database of SafeAssignment might not be comprehensive. The other Internet-based service, TurnitIn, performed reasonably well. Its detection accuracy on our test data was lower than that of SafeAssignment. Its advantage, however, is that it continually augments and uses its vast database of essays that it has collected from users of the system. EVE2 and Plagiarism-Finder failed to detect any instances of plagiarism in our test set!

Our planned future work involves using the developed data set to evaluate hermetic detection accuracy of SafeAssignment, TurnitIn and other plagiarism detection systems capable of hermetic detection. We also plan to expand the test set with other types of techniques for hiding plagiarism.

References

[1] 123HelpMe.com. http://www.123helpme.com (Accessed April 22nd, 2008).

[2] All free essays.com. http://www.allfreeessays.net (Accessed April 22nd, 2008).

[3] Canexus Inc. *EVE2- Essay Verification Engine*. http://www.canexus.com/ (Accessed April 22nd, 2008).

[4] Excite.de. http.//www.excite.de (Accessed April 29th, 2008).

[5] Mediaphor Software Entertainment AG. *Plagiarism-Finder*. http://www.m4-software.com/.

[6] Sciworth Inc. *MyDropBox*. http://www.mydropbox.com/ (Accessed April 28th, 2008).

[7] iParadigms. *TurnitIn.com. Digital assessment suite*. http://turnitin.com (Accessed April 28th, 2008).